**배열 내장함수**

forEach : 하나씩 꺼내서 적당히 처리. 첫번째 인자는 배열원소이고 두번째에 인덱스를 넣어줄수도 있다.

map : 하나씩 꺼내서 적당히 처리 후 배열로 변환.(처리결과를 리턴해서 각 배열원소로 함)

. 첫번째 인자는 배열원소이고 두번째에 인덱스를 넣어줄수도 있다.

indexOf : 원하는 항목이 몇번째 원소인지 찾아줌.

const superheroes = ['아이언맨', '캡틴 아메리카', '토르', '닥터 스트레인지'];

const index = superheroes.indexOf('토르');

console.log(index);

findIndex : 배열 안 객체라면 indexOf 로 인덱스를 찾기 어렵다. 그럴때 findIndex로 찾을 수 있다.

const todos = [

{

id: 1,

text: '자바스크립트 입문',

done: true

},

{

id: 2,

text: '함수 배우기',

done: true

},

{

id: 3,

text: '객체와 배열 배우기',

done: true

},

{

id: 4,

text: '배열 내장함수 배우기',

done: false

}

];

const index = todos.findIndex(todo => todo.id === 3);

find : findIndex하고 작동은 똑같이하는데, find는 인덱스가 아니라 객체 자체를 반환함

filter : 특정조건을 만족하는 값들만 따로 추출하여 새로운 배열을 만듬

splice : 배열에서 특정 부분을 제거함. 기존 배열이 잘라진 배열로 변화함

첫번째 파라미터로 시작 인덱스를 받고, 두번째로 몇 개를 지울지를 받음.

slice : splice와 똑같이 작용하되, 기존 배열은 건들지 않고, 새로운 배열을 만들어 반환한다.

shift : 첫번쨰 원소를 배열에서 추출해줌. 첫번째 원소를 리턴하고, 원래 배열에서 삭제함

pop : 맨 마지막 원소를 추출해줌

unshift : 배열의 첫번째에 원소를 추가함.

concat : 여러 개의 배열을 하나의 배열로 함침

const arr1 = [1,2,3];

const arr2 = [4,5,6];

const cancated = arr1.concat(arr2);

join : 배열 안의 값들을 문자열 형태로 합쳐줌

const array = [1, 2, 3, 4, 5];

console.log(array.join()); // 1,2,3,4,5

console.log(array.join(' ')); // 1 2 3 4 5

console.log(array.join(', ')); // 1, 2, 3, 4, 5

reduce : 두개의 파라미터를 받음.

첫번째는 accumulator와 currect를 파라미터로 가져와서 결과를 반환하는 콜백함수,

두번째는 reduce함수에서 사용할 초깃값

처음 accumulator는 두번째 파라미터에서 설정한 값으로 초기화됨.

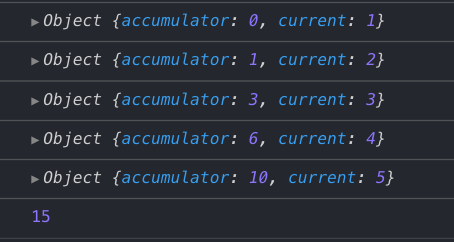
current로 배열의 원소를 하나씩 받아와서 적절히 처리해준다.

let sum = numbers.reduce((accumulator, current) => {

console.log({ accumulator, current });

return accumulator + current;

}, 0);



**객체 내장함수**

Object.entries : [[키, 값], [키,값]] 형태의 배열로 반환

Object.keys : [키, 키, 키] 형태의 배열로 변환

Object.values : [값, 값, 값] 형태의 배열로 변환